

Mobile Application Development

Prof. Fernando Pinéo

Navegação entre telas

Navegação entre telas



Há 03 formas de navegação entre as telas:

Stack Navigation

Bottom Tabs

Drawer Navigation

Stack Navigation

FIAP

reactnavigation.org/docs/stack-navigator

React Navigation

Web support

Server rendering

Screen tracking

Themes

State persistence

Static and dynamic APIs

Testing with Jest

Configuring TypeScript

Troubleshooting

Upgrading from 6.x

Navigators

Stack

Native Stack

Bottom Tabs

Drawer

Material Top Tabs

Libraries

Developer tools

Elements

Tab View

Drawer Layout

7.x Docs Blog

Stack Navigator provides a way for your app to transition between screens where each new screen is placed on top of a stack.

By default the stack navigator is configured to have the familiar iOS and Android look & feel: new screens slide in from the right on iOS, use OS default animation on Android. But the animations can be customized to match your needs.

12:30

← Feed

Navigate to albumGo back

What's on your mind?


Joke bot

If you aren't impressed with the picture of the first Black Hole, you clearly don't understand the gravity of the situation.

♡💬🔗

Joke bot

I went to the zoo and I saw a baguette in a cage. I asked the zookeeper about it and he said it was bread in captivity.



♡💬🔗

Joke bot

Why didn't 4 ask 5 out? Because he was 2*.

♡💬🔗

Network Provider 00:24

← Home Feed

Navigate to albumGo back

What's on your mind?


Joke bot

If you aren't impressed with the picture of the first Black Hole, you clearly don't understand the gravity of the situation.

♡💬🔗

Joke bot


I went to the zoo and I saw a baguette in a cage. I asked the zookeeper about it and he said it was bread in captivity.



Bottom Navigation

FIAP

reactnavigation.org/docs/bottom-tab-navigator

 **React Navigation**

Web support

Server rendering

Screen tracking

Themes

State persistence

Static and dynamic APIs

Testing with Jest

Configuring TypeScript

Troubleshooting

Upgrading from 6.x

Navigators

Stack

Native Stack

Bottom Tabs

Drawer

Material Top Tabs

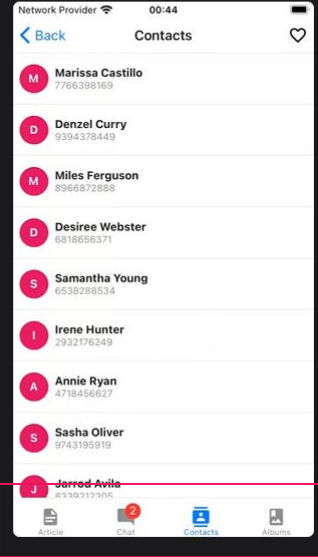
Libraries

Developer tools

7.x ▾ Docs Blog Help ▾

Bottom Tabs Navigator

A simple tab bar on the bottom of the screen that lets you switch between different routes. Routes are lazily initialized -- their screen components are not mounted until they are first focused.



Drawer Navigation

FIAP

The screenshot shows the React Navigation documentation page for the Drawer Navigator. The page is dark-themed with a light blue sidebar on the left and a light blue header. The sidebar contains a list of navigation options, with 'Drawer' highlighted under the 'Navigators' section. The main content area features a 'Version: 7.x' badge, the title 'Drawer Navigator', and a description: 'Drawer Navigator renders a navigation drawer on the side of the screen which can be opened and closed via gestures.' Below the text is a mobile app mockup showing a drawer with four items: 'Inbox', 'Outbox' (highlighted with a blue bar), 'Favorites', and 'Trash'. The top of the mockup shows the time '12:43' and status bar icons. The right sidebar contains a list of links: 'Installation', 'Usage', 'API Definition', 'Props', 'Options', 'Header', 'Events', 'Hooks', and 'Nesting'.

reactnavigation.org/docs/drawer-navigator

React Navigation

7.x ▾ Docs Blog Help ▾

Version: 7.x

Drawer Navigator

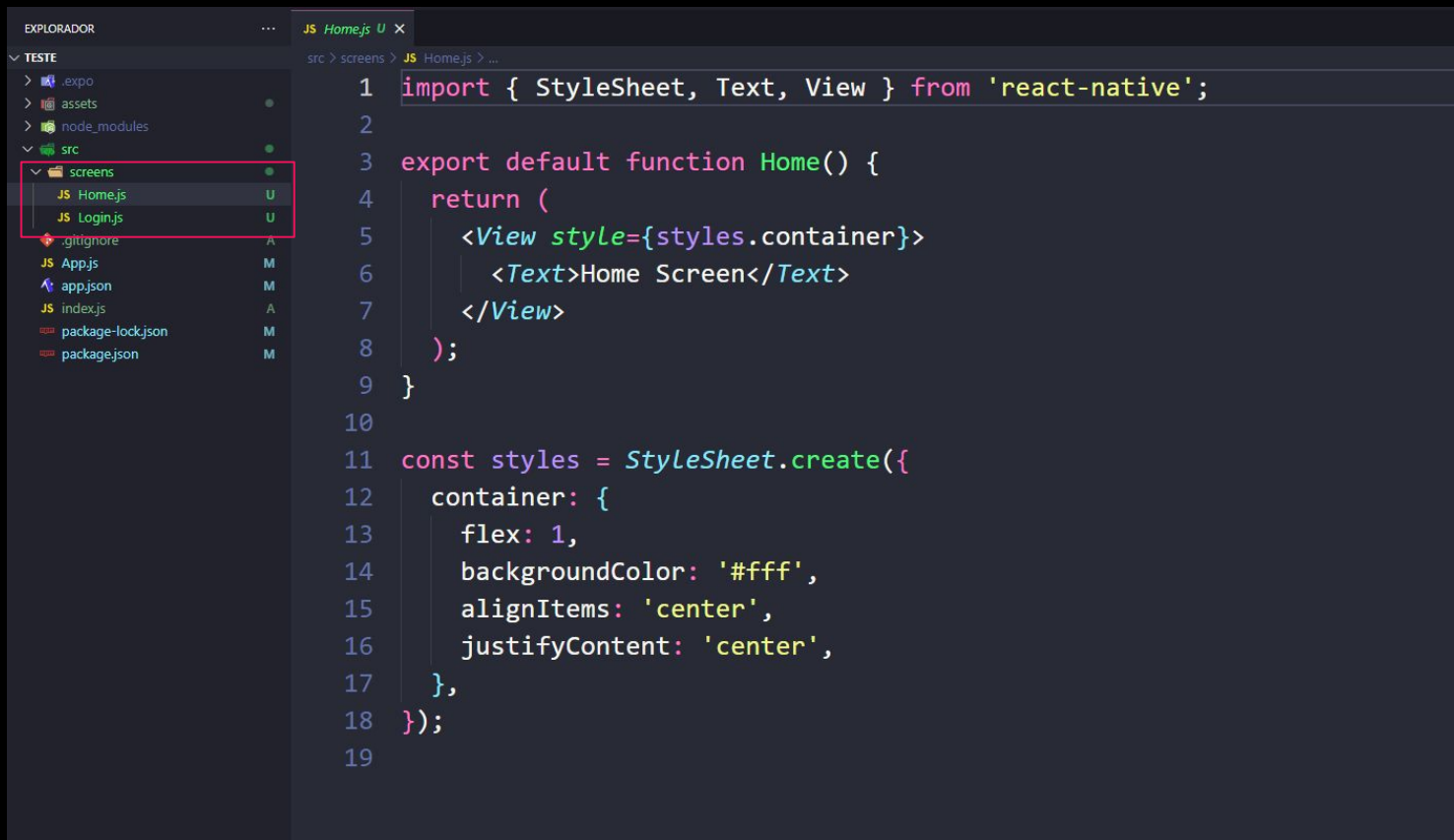
Drawer Navigator renders a navigation drawer on the side of the screen which can be opened and closed via gestures.

12:43

- Inbox
- Outbox
- Favorites
- Trash

Installation
Usage
API Definition
Props
Options
Header
Events
Hooks
Nesting

Criando a Tela Home e Tela Login

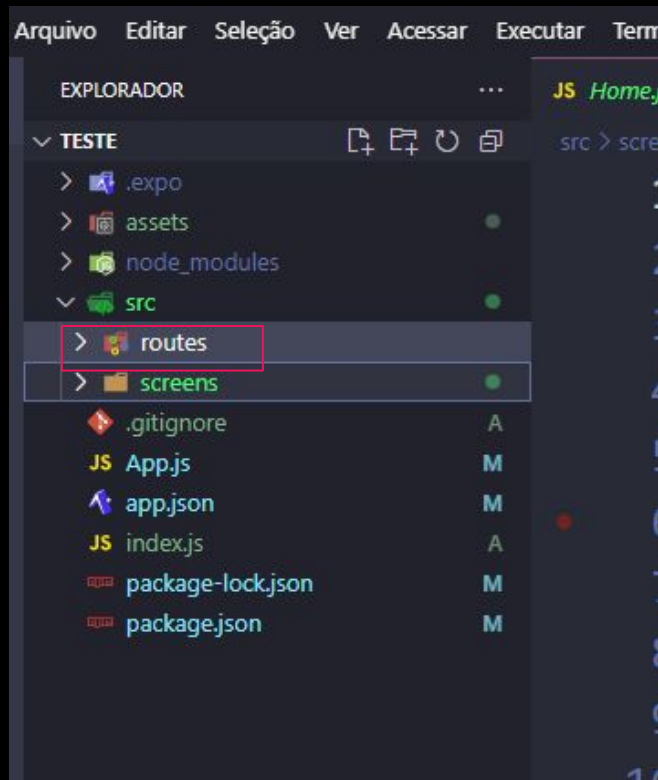


The screenshot shows a code editor with a dark theme. On the left, the 'EXPLORADOR' (Explorer) panel is open, showing a file tree. The 'src' folder is expanded, and the 'screens' subfolder is highlighted with a red box. Inside 'screens', there are two files: 'Home.js' and 'Login.js', both marked with a 'U' (Unsaved). Below 'screens' are other files like '.gitignore', 'App.js', 'app.json', 'index.js', 'package-lock.json', and 'package.json'. The main editor area shows the code for 'Home.js'. The code imports 'StyleSheet', 'Text', and 'View' from 'react-native'. It defines a default function 'Home()' that returns a JSX element. The JSX element consists of a 'View' with a style object 'styles.container', containing a 'Text' component with the text 'Home Screen'. The 'styles' object is created using 'StyleSheet.create()' with properties for 'flex', 'backgroundColor', 'alignItems', and 'justifyContent'. The code is as follows:

```
1 import { StyleSheet, Text, View } from 'react-native';
2
3 export default function Home() {
4   return (
5     <View style={styles.container}>
6       <Text>Home Screen</Text>
7     </View>
8   );
9 }
10
11 const styles = StyleSheet.create({
12   container: {
13     flex: 1,
14     backgroundColor: '#fff',
15     alignItems: 'center',
16     justifyContent: 'center',
17   },
18 });
19
```

Criando a pasta Routes(rotas de navegação do nosso app).

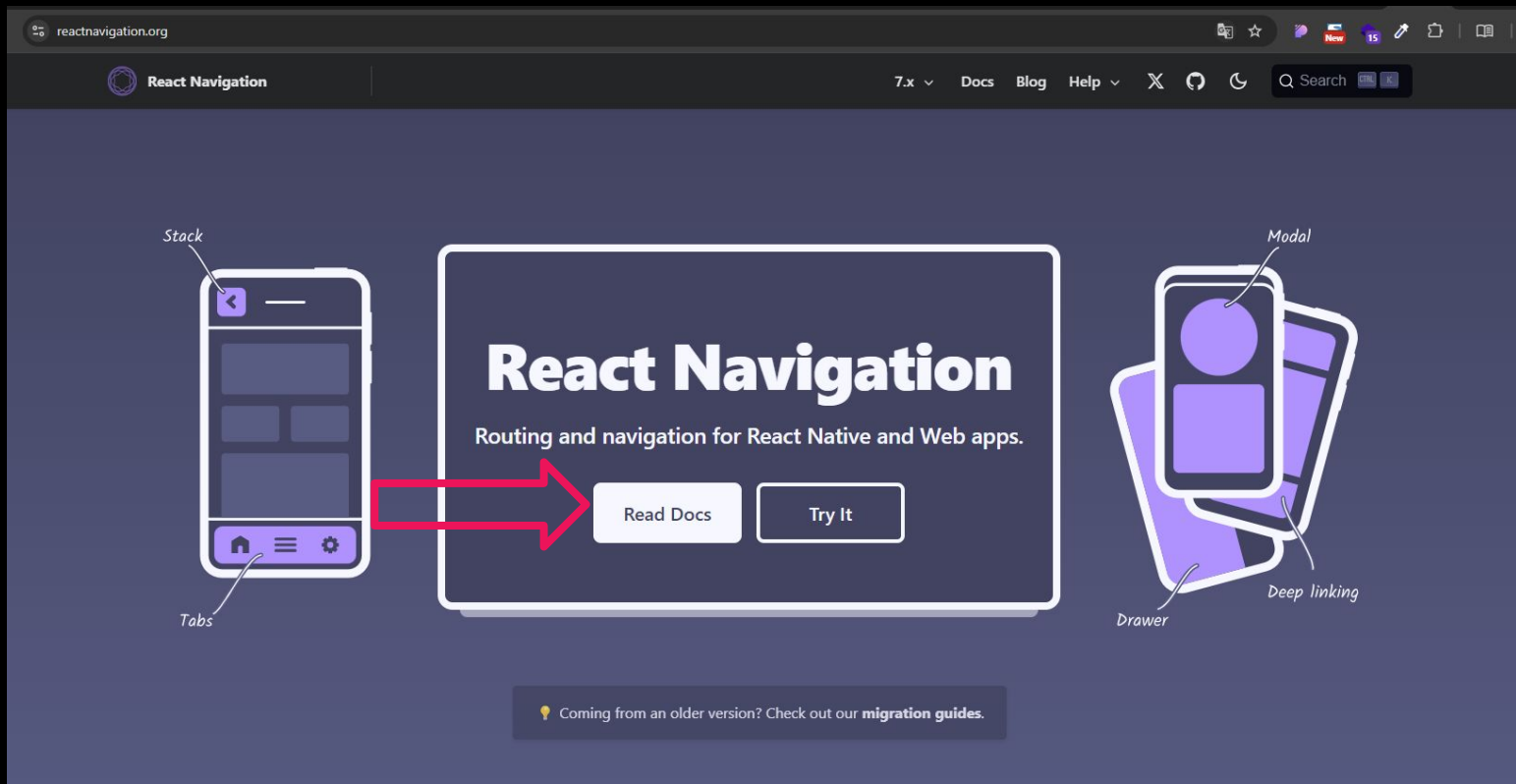
FIAP



Pesquisa o React Navigation

FIAP

<https://reactnavigation.org/>



Instalação do Core de navegação

A screenshot of the React Navigation documentation website. The browser's address bar shows 'reactnavigation.org/docs/getting-started'. The website has a dark theme. On the left, a sidebar menu under the heading 'Fundamentals' lists several topics, with 'Getting started' highlighted by a red box. The main content area is titled 'Installation' and contains the text 'Install the required packages in your React Native project:'. Below this text are three tabs: 'npm', 'Yarn', and 'pnpm'. The 'npm' tab is selected and highlighted by a red box. Under the 'npm' tab, the command 'npm install @react-navigation/native' is displayed in a code block, also highlighted by a red box. The rest of the page content is partially visible below the code block.

reactnavigation.org/docs/getting-started

React Navigation

7.x ▾ Docs Blog Help ▾

Fundamentals ▾

- Getting started
- Hello React Navigation
- Moving between screens
- Passing parameters to routes
- Configuring the header bar
- Header buttons
- Nesting navigators
- Navigation lifecycle

Installation

Install the required packages in your React Native project:

npm Yarn pnpm

```
npm install @react-navigation/native
```

React Navigation is made up of some core utilities and those are then used by navigators to create the navigation structure in your app. Don't worry too much about this for now, it'll become clear soon enough! To frontload the installation work, let's also install and configure dependencies used

Instalando as dependências de navegação



The screenshot shows the React Navigation documentation website. The left sidebar contains a navigation menu with 'Fundamentals' expanded, highlighting 'Getting started'. The main content area is titled 'Installing dependencies into an Expo managed project' and includes a code block for the command to install dependencies. Below this, it explains that compatible versions will be installed and provides a link to 'Hello React Navigation'. A second section, 'Installing dependencies into a bare React Native project', is partially visible at the bottom.

reactnavigation.org/docs/getting-started

React Navigation

7.x ▾ Docs Blog Help ▾

Fundamentals

- Getting started
- Hello React Navigation
- Moving between screens
- Passing parameters to routes
- Configuring the header bar
- Header buttons
- Nesting navigators
- Navigation lifecycle
- Next steps

Guides

Installing dependencies into an Expo managed project

In your project directory, run:

```
npx expo install react-native-screens react-native-safe-area-context
```

This will install versions of these libraries that are compatible.

You can now continue to ["Hello React Navigation"](#) to start writing some code.

Installing dependencies into a bare React Native project

In your project directory, run:

npm Yarn pnpm

Instalando a navegação Bottom-tabs



reactnavigation.org/docs/bottom-tab-navigator

React Navigation

Navigation kit

Deep linking

Configuring links

Web support

Server rendering

Screen tracking

Themes

State persistence

Static and dynamic APIs

Testing with Jest

Configuring TypeScript

Troubleshooting

Upgrading from 6.x

Navigators

Stack

Native Stack

Bottom Tabs

Drawer

Material Top Tabs

7.x Docs Blog Help

Gandalf
1st Jan 2025

What is Lorem Ipsum?

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Article

Chat

Contacts

Albums

Installation

To use this navigator, ensure that you have `@react-navigation/native` and its dependencies (follow this guide), then install `@react-navigation/bottom-tabs`:

npm

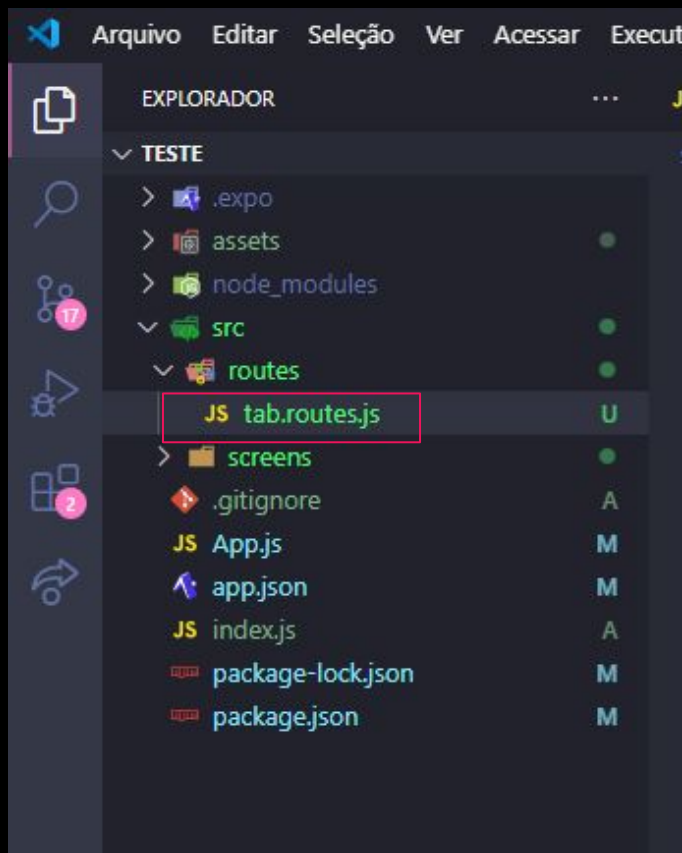
Yarn

pnpm

```
npm install @react-navigation/bottom-tabs
```

Criando arq de navegação Bottom

FIAP



Arq tab.routes, configurando as rotas



The screenshot shows a code editor with a dark theme. On the left, the 'EXPLORADOR' (Explorer) panel shows a file tree with folders like 'assets', 'node_modules', 'src', and 'screens'. The 'src' folder is expanded, showing 'routes' and 'screens'. The 'routes' folder is selected, and the file 'tab.routes.js' is open. The main editor area shows the code for 'tab.routes.js'. The code imports 'createBottomTabNavigator' from '@react-navigation/bottom-tabs', 'Home' from '../screens/Home', and 'Login' from '../screens/Login'. It then defines a 'const Tab = createBottomTabNavigator()' and an 'export default function TabRoutes()' which returns a JSX element representing a bottom tab navigator with two screens: 'home' and 'login'.

```
1 import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";
2 import Home from "../screens/Home";
3 import Login from "../screens/Login"
4
5
6 const Tab = createBottomTabNavigator()
7
8 export default function TabRoutes(){
9   return(
10     <Tab.Navigator>
11       <Tab.Screen name="home" component={Home}/>
12       <Tab.Screen name="login" component={Login}/>
13     </Tab.Navigator>
14   )
15 }
```

Criando o arq Index com a estratégia de rotas $F|/\backslash/P$ escolhida

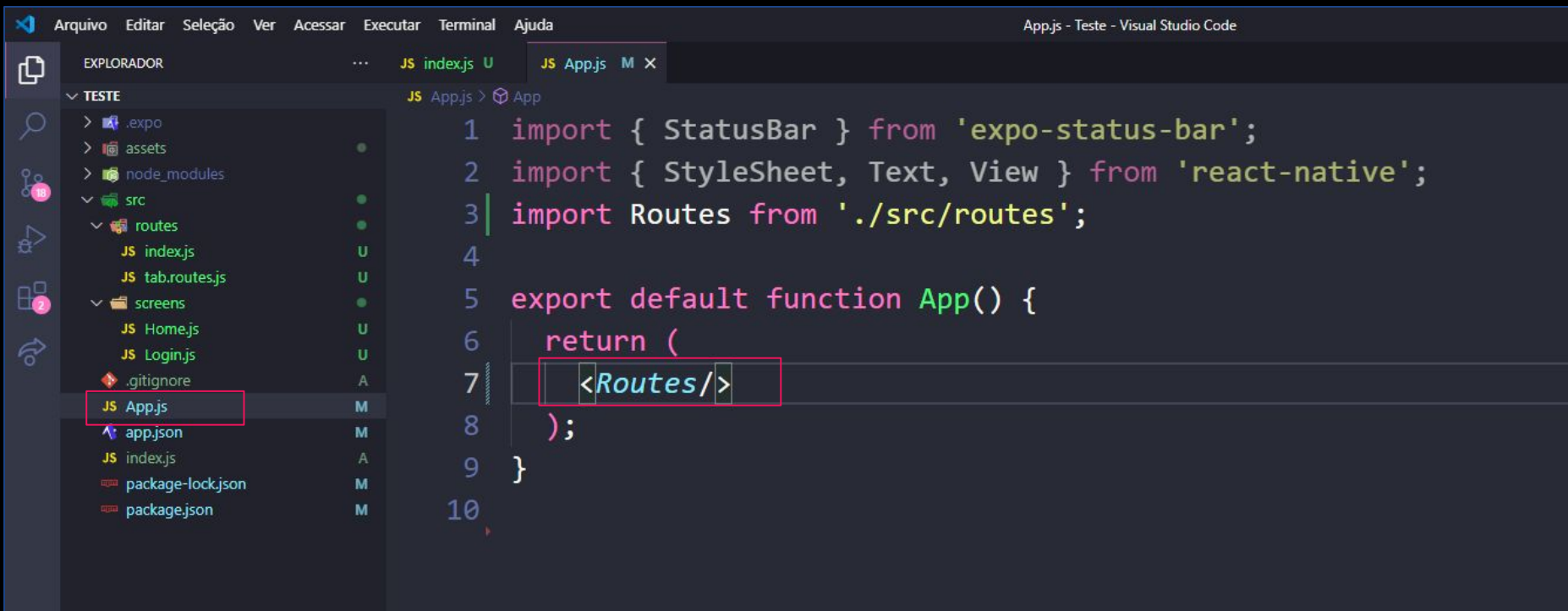


The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure under the 'TESTE' folder. The 'routes' folder is expanded, and 'index.js' is selected and highlighted with a red box. The main editor area shows the content of 'index.js', which is also highlighted with a red box. The code defines a 'Routes' function that returns a JSX element containing a 'NavigationContainer' and a 'TabRoutes' component. The 'TabRoutes' component is highlighted with a red box.

```
1 import { NavigationContainer } from "@react-navigation/native";
2 import TabRoutes from "./tab.routes";
3
4 export default function Routes(){
5   return(
6     <NavigationContainer>
7     <TabRoutes/>
8   </NavigationContainer>
9 )
10 }
```

No App.js importo o meu Routes

FIA/P



```
Arquivo  Editar  Seleção  Ver  Acessar  Executar  Terminal  Ajuda  App.js - Teste - Visual Studio Code

EXPLORADOR  JS index.js U  JS App.js M x

TESTE  JS App.js > App

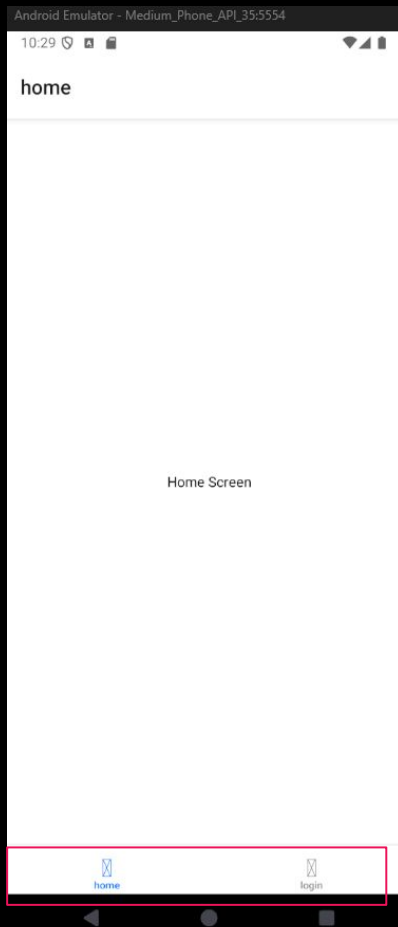
> .expo
> assets
> node_modules
└─ src
   └─ routes
      JS index.js U
      JS tab.routes.js U
   └─ screens
      JS Home.js U
      JS Login.js U
      .gitignore A
  JS App.js M
  app.json M
  JS index.js A
  package-lock.json M
  package.json M

1 import { StatusBar } from 'expo-status-bar';
2 import { StyleSheet, Text, View } from 'react-native';
3 import Routes from './src/routes';
4
5 export default function App() {
6   return (
7     <Routes/>
8   );
9 }
10
```


Recapitulando..

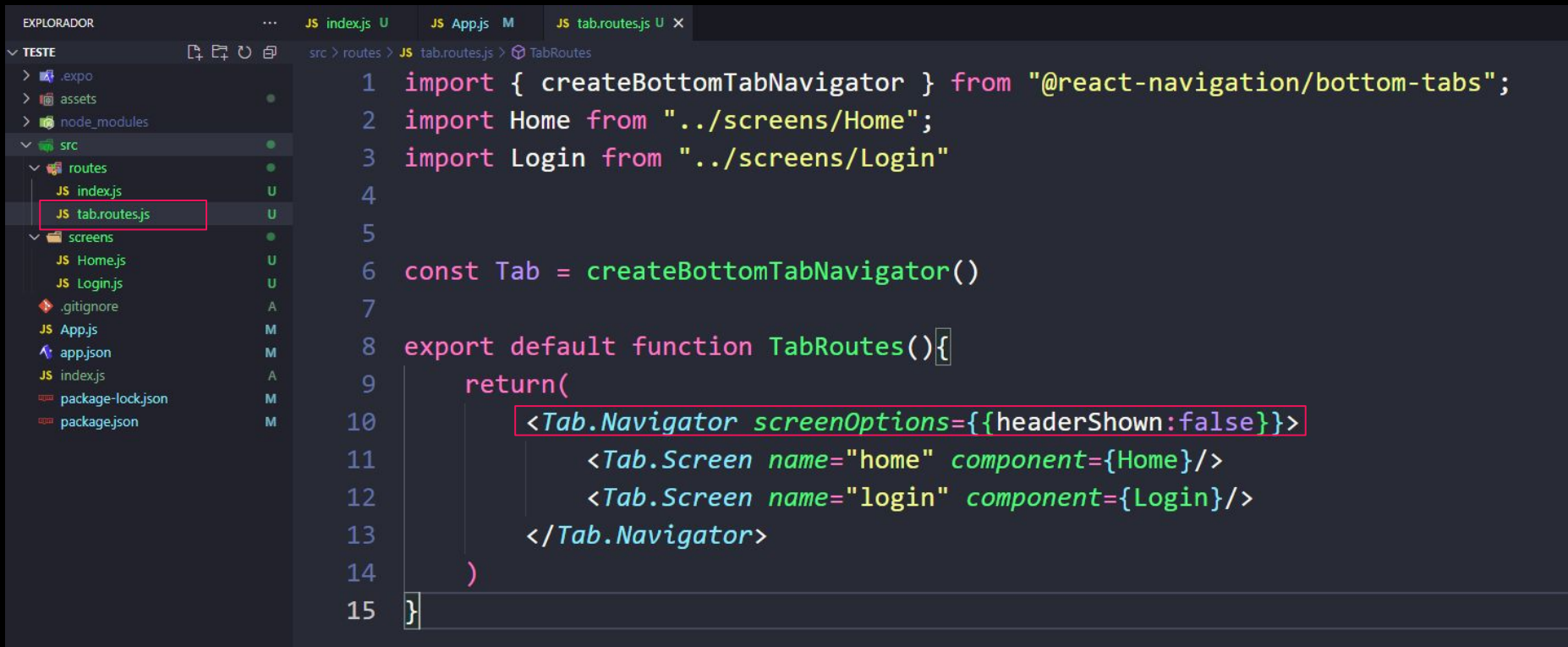
Resultado até aqui

FIAP



Tirando o cabeçalho default

FIAP



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like .expo, assets, node_modules, src, routes, and screens. The file JS tab.routes.js is selected and highlighted with a red box. The code editor shows the content of JS tab.routes.js, which is a TabRoutes component. The code is as follows:

```
1 import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";
2 import Home from "../screens/Home";
3 import Login from "../screens/Login"
4
5
6 const Tab = createBottomTabNavigator()
7
8 export default function TabRoutes(){
9   return(
10     <Tab.Navigator screenOptions={{headerShown:false}}>
11       <Tab.Screen name="home" component={Home}/>
12       <Tab.Screen name="login" component={Login}/>
13     </Tab.Navigator>
14   )
15 }
```

The code is written in a dark-themed editor. The file explorer on the left shows a project structure with folders like .expo, assets, node_modules, src, routes, and screens. The file JS tab.routes.js is selected and highlighted with a red box. The code editor shows the content of JS tab.routes.js, which is a TabRoutes component. The code is as follows:

Colocando Ícones e aumentando os ícones

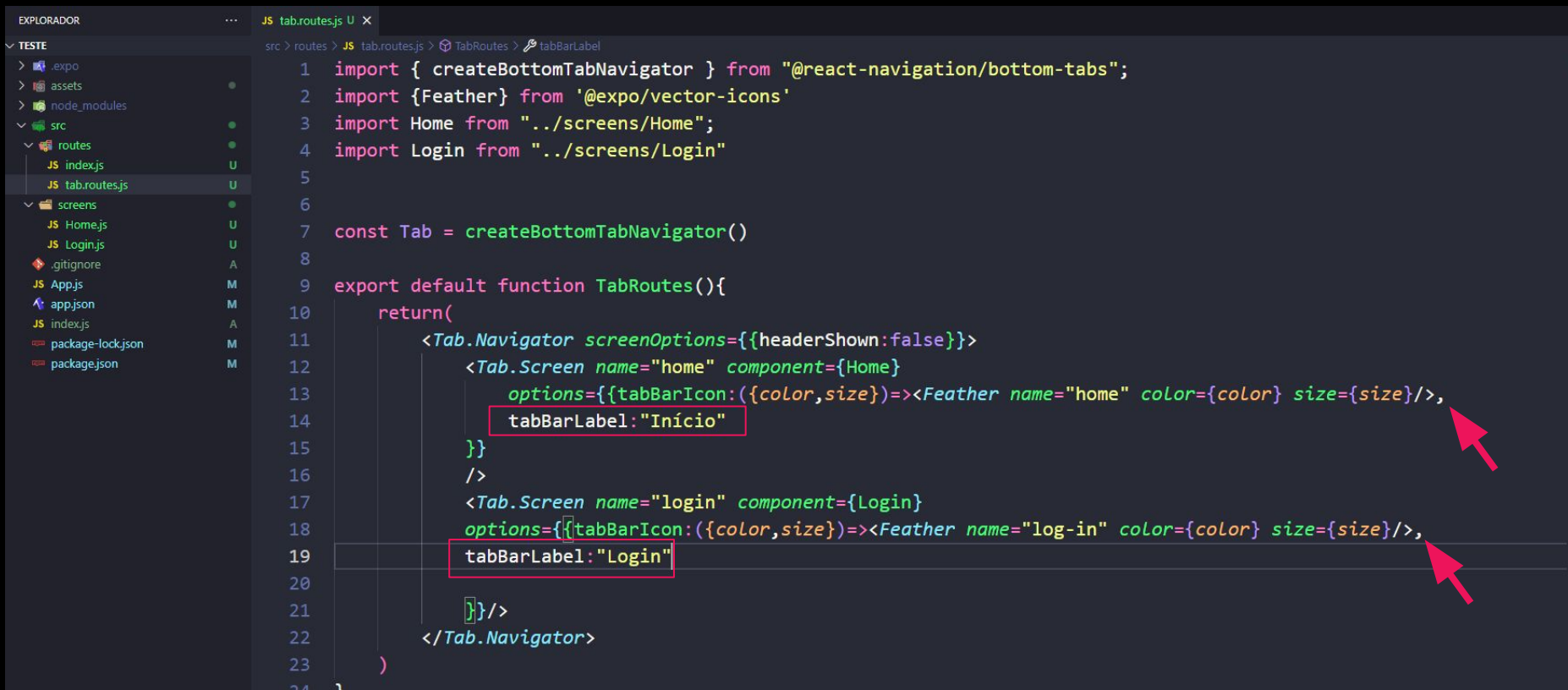
FILIP



```
EXPLORADOR
...
▼ TESTE
  > .expo
  > assets
  > node_modules
  ▼ src
    ▼ routes
      JS index.js U
      JS tab.routes.js U
    ▼ screens
      JS Home.js U
      JS Login.js U
    .gitignore A
    JS App.js M
    app.json M
    JS index.js A
    package-lock.json M
    package.json M

JS tab.routes.js X
src > routes > JS tab.routes.js > TabRoutes
1 import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";
2 import { Feather } from '@expo/vector-icons'
3 import Home from "../screens/Home";
4 import Login from "../screens/Login"
5
6
7 const Tab = createBottomTabNavigator()
8
9 export default function TabRoutes(){
10   return(
11     <Tab.Navigator screenOptions={{headerShown:false}}>
12       <Tab.Screen name="home" component={Home}
13         options={{tabBarIcon:({color,size})=><Feather name="home" color={color} size={size}/>}}
14       />
15       <Tab.Screen name="login" component={Login}
16         options={{tabBarIcon:({color,size})=><Feather name="log-in" color={color} size={size}/>}}
17     </Tab.Navigator>
18   )
19 }
```

Alterando o texto do menu do bottom tabs F|/\P



The image shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like .expo, assets, node_modules, src, routes, and screens. The code editor displays the file `tab.routes.js` with the following code:

```
1 import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";
2 import { Feather } from '@expo/vector-icons'
3 import Home from "../screens/Home";
4 import Login from "../screens/Login"
5
6
7 const Tab = createBottomTabNavigator()
8
9 export default function TabRoutes(){
10   return(
11     <Tab.Navigator screenOptions={{headerShown:false}}>
12       <Tab.Screen name="home" component={Home}
13         options={{tabBarIcon:({color,size})=><Feather name="home" color={color} size={size}/>,
14           tabBarLabel:"Início"
15         }}
16       />
17       <Tab.Screen name="login" component={Login}
18         options={{tabBarIcon:({color,size})=><Feather name="log-in" color={color} size={size}/>,
19           tabBarLabel:"Login"
20         }}
21       />
22     </Tab.Navigator>
23   )
24 }
```

Two red arrows point to the `tabBarLabel` property in the options object for the "home" and "login" screens, indicating the text being modified.

Adicionando nova forma de navegação

Drawer navigation

Instalando o Drawer Navigation no projeto

A screenshot of the React Navigation documentation website, specifically the page for the Drawer Navigator. The page has a dark theme. On the left is a sidebar with a 'React Navigation' header and a list of links: Web support, Server rendering, Screen tracking, Themes, State persistence, Static and dynamic APIs, Testing with Jest, Configuring TypeScript, Troubleshooting, Upgrading from 6.x, Navigators (with a dropdown arrow), Stack, Native Stack, Bottom Tabs, Drawer (highlighted with a red box), and Material Top Tabs. The main content area has a header with '7.x', 'Docs', 'Blog', 'Help', and social media icons. The text explains that the Drawer Navigator wraps 'react-native-drawer-layout' and provides instructions for installation. It lists three package managers: npm, Yarn, and pnpm. A code block shows the command 'npm install @react-navigation/drawer'. Below this, it states that additional libraries are required and lists the first step: installing 'react-native-gesture-handler' and 'react-native-reanimated'. Another code block shows the command 'npx expo install react-native-gesture-handler react-native-reanimated'.

reactnavigation.org/docs/drawer-navigator

React Navigation

7.x ▾ Docs Blog Help ▾ X

Web support

Server rendering

Screen tracking

Themes

State persistence

Static and dynamic APIs

Testing with Jest

Configuring TypeScript

Troubleshooting

Upgrading from 6.x

Navigators ▾

Stack

Native Stack

Bottom Tabs

Drawer

Material Top Tabs

This wraps `react-native-drawer-layout`. If you want to use the drawer without React Navigation integration, use the library directly instead.

Installation

To use this navigator, ensure that you have `@react-navigation/native` and its dependencies (follow this guide), then install `@react-navigation/drawer`:

npm Yarn pnpm

```
npm install @react-navigation/drawer
```

Then, you need to install and configure the libraries that are required by the drawer navigator:

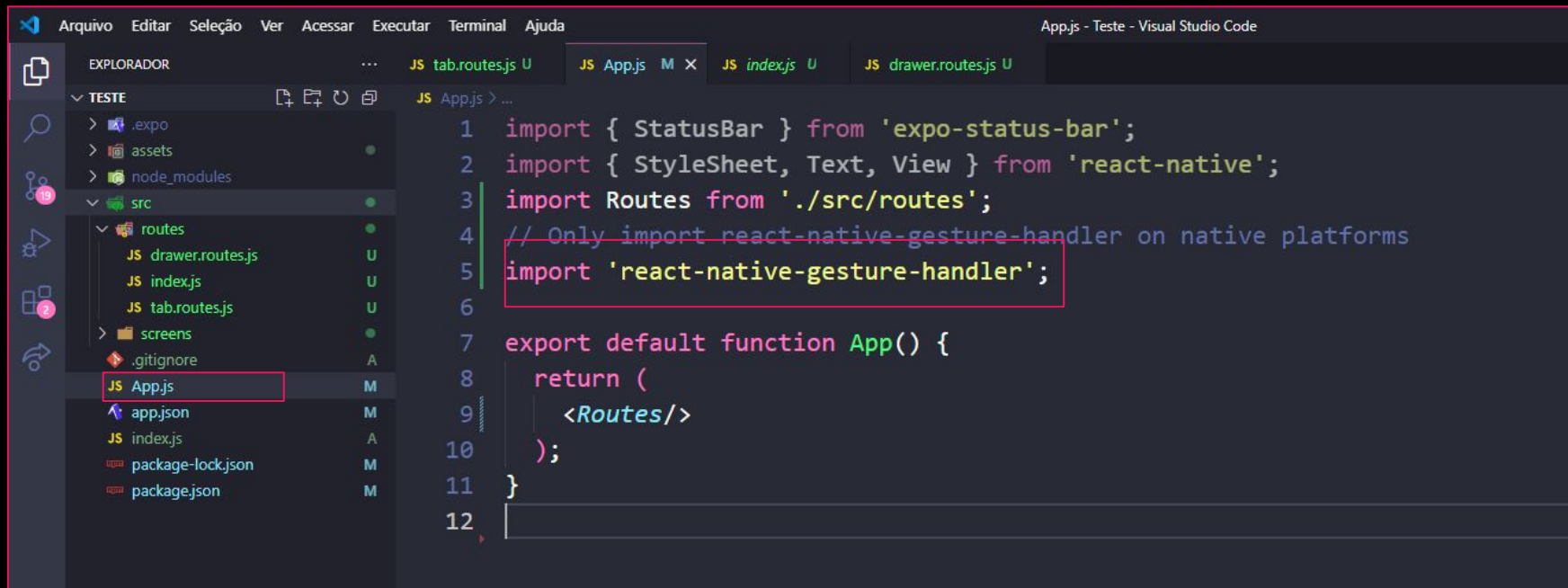
1. First, install `react-native-gesture-handler` and `react-native-reanimated` (at least version 2 or 3).

If you have a Expo managed project, in your project directory, run:

```
npx expo install react-native-gesture-handler react-native-reanimated
```

Importando gesture-handler

FIAP

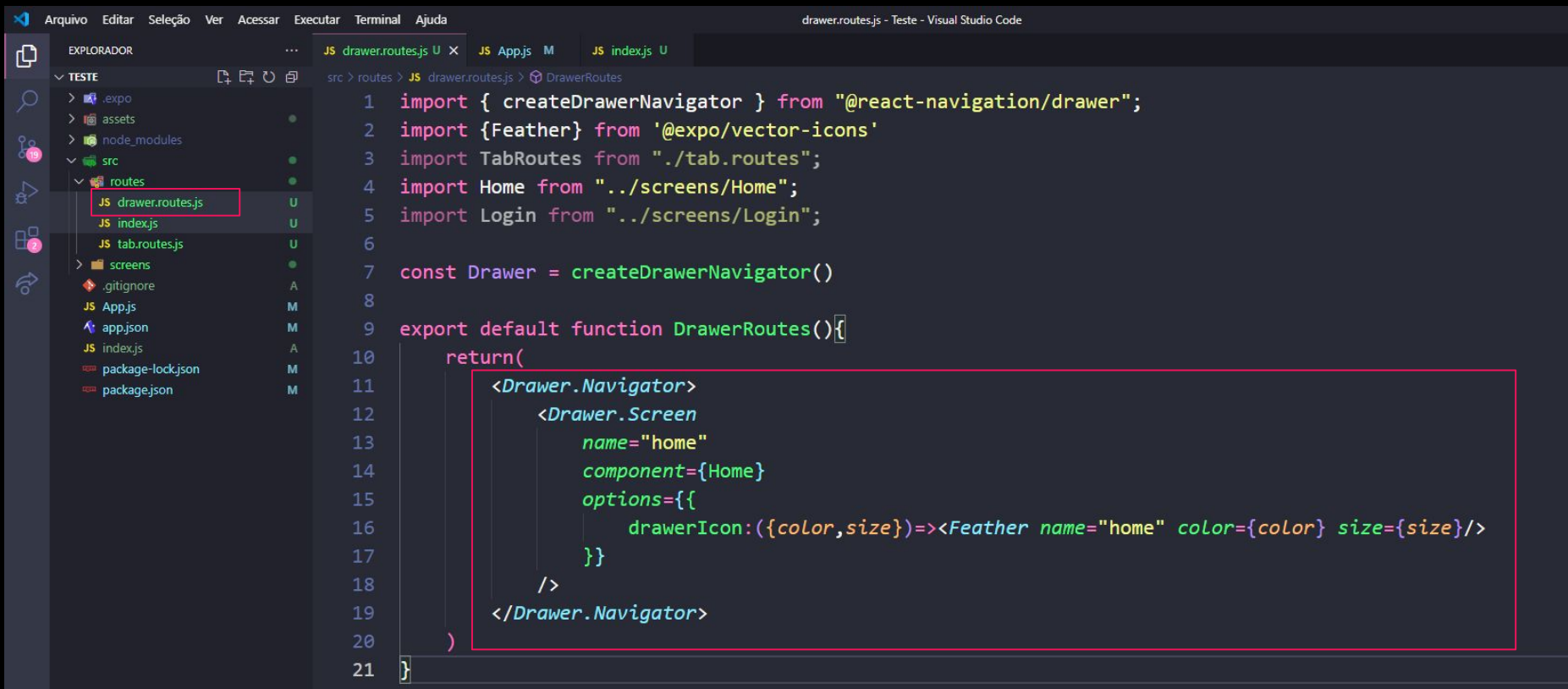


The image shows a screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left displays the project structure, with the file `App.js` selected and highlighted by a red rectangle. The main editor area shows the content of `App.js`. The code includes imports for `StatusBar` from `'expo-status-bar'`, `StyleSheet`, `Text`, and `View` from `'react-native'`, and `Routes` from `'./src/routes'`. A comment indicates that `react-native-gesture-handler` should only be imported on native platforms. This import statement is highlighted by a red rectangle. The `App` function is defined as the default export, returning a `<Routes/>` component.

```
1 import { StatusBar } from 'expo-status-bar';
2 import { StyleSheet, Text, View } from 'react-native';
3 import Routes from './src/routes';
4 // Only import react-native-gesture-handler on native platforms
5 import 'react-native-gesture-handler';
6
7 export default function App() {
8   return (
9     <Routes/>
10  );
11 }
12
```

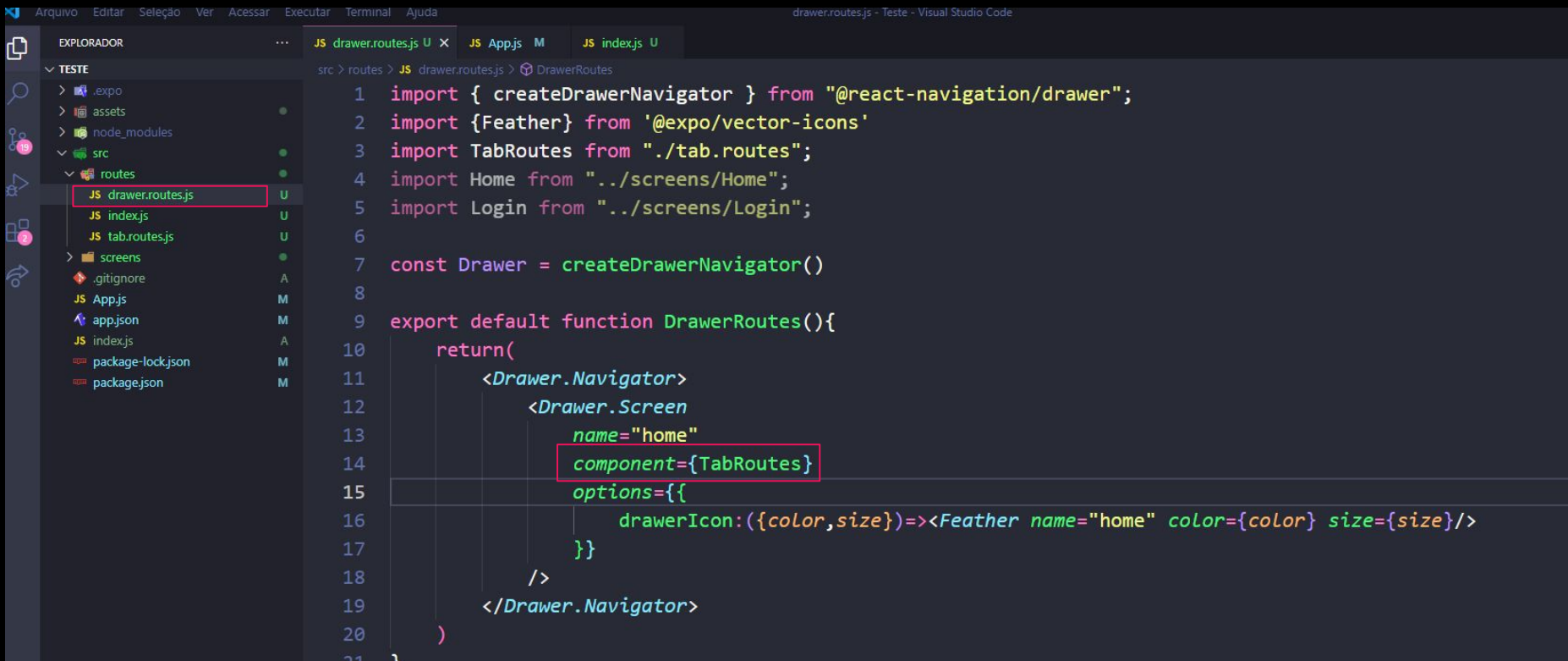

Nosso arquivo drawer.routes.js

FIAP



```
1 import { createDrawerNavigator } from "@react-navigation/drawer";
2 import { Feather } from '@expo/vector-icons'
3 import TabRoutes from "../tab.routes";
4 import Home from "../screens/Home";
5 import Login from "../screens/Login";
6
7 const Drawer = createDrawerNavigator()
8
9 export default function DrawerRoutes(){
10   return(
11     <Drawer.Navigator>
12       <Drawer.Screen
13         name="home"
14         component={Home}
15         options={{
16           drawerIcon: ({color,size})=><Feather name="home" color={color} size={size}/>
17         }}
18       />
19     </Drawer.Navigator>
20   )
21 }
```

Caso eu queira os dois tipo de navegação



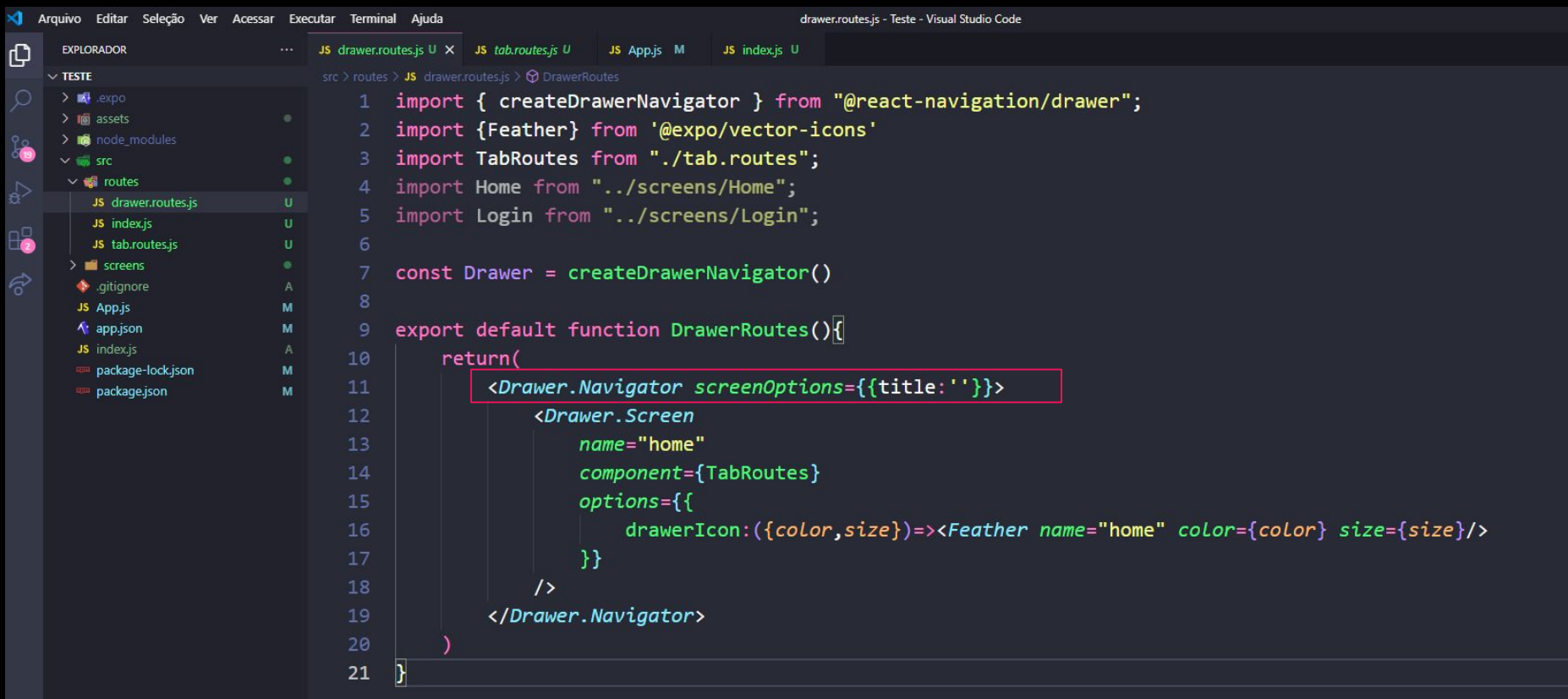
```
Arquivo  Editar  Seleção  Ver  Acessar  Executar  Terminal  Ajuda  drawer.routes.js - Teste - Visual Studio Code

EXPLORADOR
... JS drawer.routes.js U X JS App.js M JS index.js U
▼ TESTE
  > .expo
  > assets
  > node_modules
  ▼ src
    ▼ routes
      JS drawer.routes.js U
      JS index.js U
      JS tab.routes.js U
    > screens
  > .gitignore A
  JS App.js M
  app.json M
  JS index.js A
  package-lock.json M
  package.json M

src > routes > JS drawer.routes.js > DrawerRoutes
1  import { createDrawerNavigator } from "@react-navigation/drawer";
2  import { Feather } from '@expo/vector-icons'
3  import TabRoutes from "../tab.routes";
4  import Home from "../screens/Home";
5  import Login from "../screens/Login";
6
7  const Drawer = createDrawerNavigator()
8
9  export default function DrawerRoutes(){
10    return(
11      <Drawer.Navigator>
12        <Drawer.Screen
13          name="home"
14          component={TabRoutes}
15          options={{
16            drawerIcon: ({color,size})=><Feather name="home" color={color} size={size}/>
17          }}
18        />
19      </Drawer.Navigator>
20    )
21  }
```

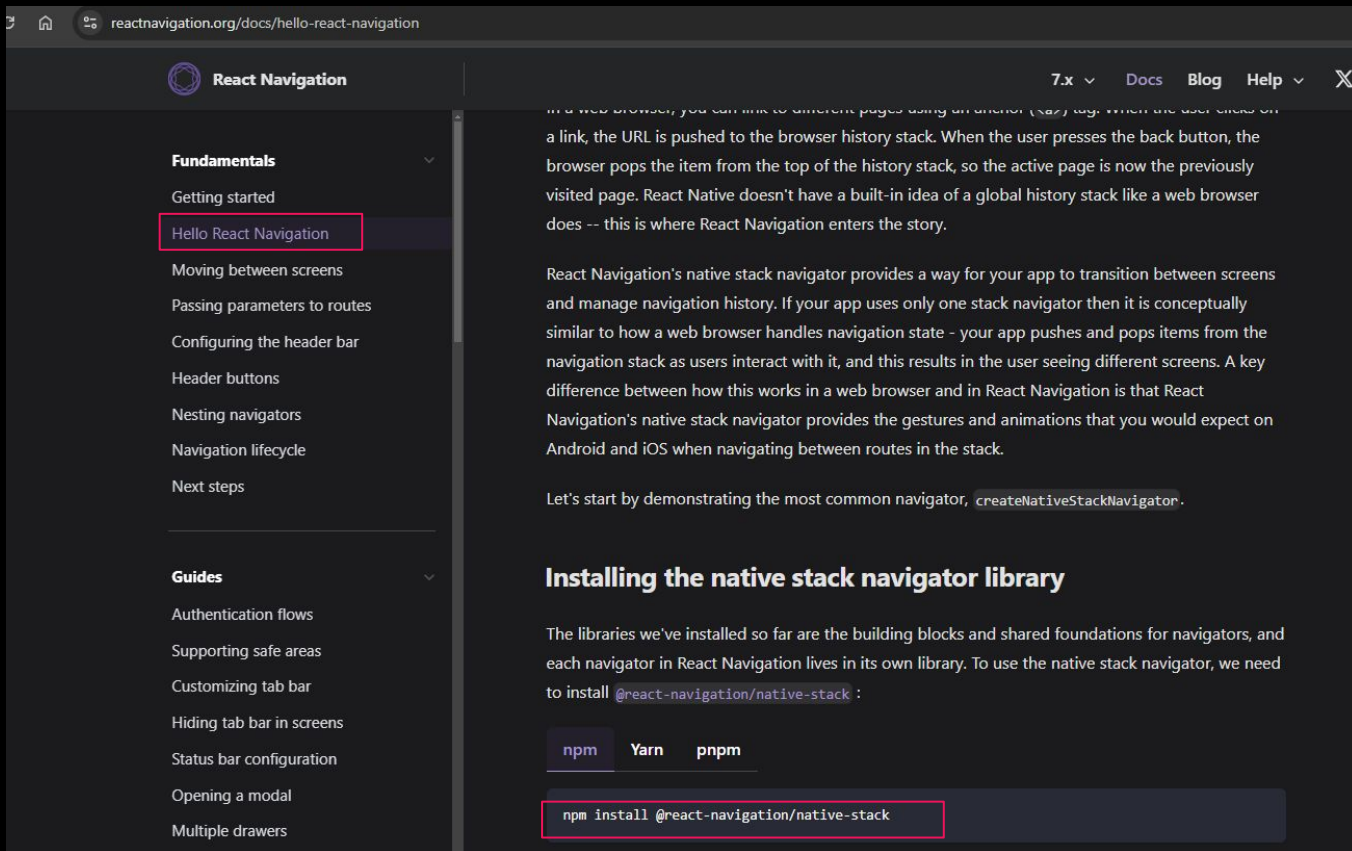
Escondendo o título do drawer

FIAP



```
1 import { createDrawerNavigator } from "@react-navigation/drawer";
2 import { Feather } from '@expo/vector-icons'
3 import TabRoutes from "../tab.routes";
4 import Home from "../screens/Home";
5 import Login from "../screens/Login";
6
7 const Drawer = createDrawerNavigator()
8
9 export default function DrawerRoutes(){
10   return(
11     <Drawer.Navigator screenOptions={{title:''}}>
12       <Drawer.Screen
13         name="home"
14         component={TabRoutes}
15         options={{
16           drawerIcon: ({color,size})=><Feather name="home" color={color} size={size}/>
17         }}
18       />
19     </Drawer.Navigator>
20   )
21 }
```

Navegação Stack



The screenshot shows the React Navigation documentation website. The left sidebar contains a navigation menu with two main sections: 'Fundamentals' and 'Guides'. Under 'Fundamentals', the 'Hello React Navigation' link is highlighted with a red box. The main content area on the right contains text explaining how navigation works in a web browser versus React Native, and how React Navigation's native stack navigator provides a way to transition between screens. Below this, there is a section titled 'Installing the native stack navigator library' which includes a code block for installing the library using npm, Yarn, or pnpm. The 'npm' option is selected, and the command 'npm install @react-navigation/native-stack' is highlighted with a red box.

reactnavigation.org/docs/hello-react-navigation

React Navigation

7.x Docs Blog Help

Fundamentals

- Getting started
- Hello React Navigation**
- Moving between screens
- Passing parameters to routes
- Configuring the header bar
- Header buttons
- Nesting navigators
- Navigation lifecycle
- Next steps

Guides

- Authentication flows
- Supporting safe areas
- Customizing tab bar
- Hiding tab bar in screens
- Status bar configuration
- Opening a modal
- Multiple drawers

in a web browser, you can link to different pages using an anchor (`<a>`) tag. When the user clicks on a link, the URL is pushed to the browser history stack. When the user presses the back button, the browser pops the item from the top of the history stack, so the active page is now the previously visited page. React Native doesn't have a built-in idea of a global history stack like a web browser does -- this is where React Navigation enters the story.

React Navigation's native stack navigator provides a way for your app to transition between screens and manage navigation history. If your app uses only one stack navigator then it is conceptually similar to how a web browser handles navigation state - your app pushes and pops items from the navigation stack as users interact with it, and this results in the user seeing different screens. A key difference between how this works in a web browser and in React Navigation is that React Navigation's native stack navigator provides the gestures and animations that you would expect on Android and iOS when navigating between routes in the stack.

Let's start by demonstrating the most common navigator, `createNativeStackNavigator`.

Installing the native stack navigator library

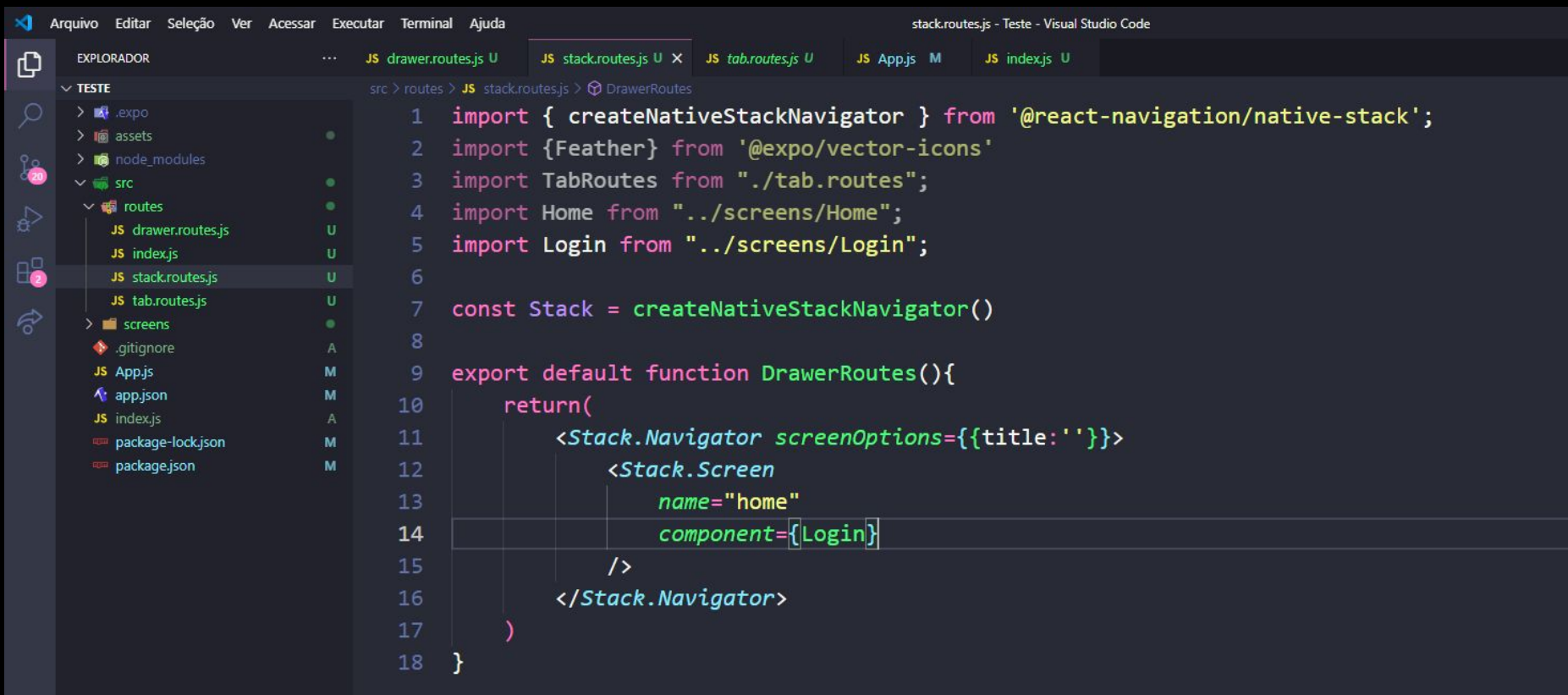
The libraries we've installed so far are the building blocks and shared foundations for navigators, and each navigator in React Navigation lives in its own library. To use the native stack navigator, we need to install [@react-navigation/native-stack](#) :

npm Yarn pnpm

```
npm install @react-navigation/native-stack
```

Arq stack.routes.js

FIAP

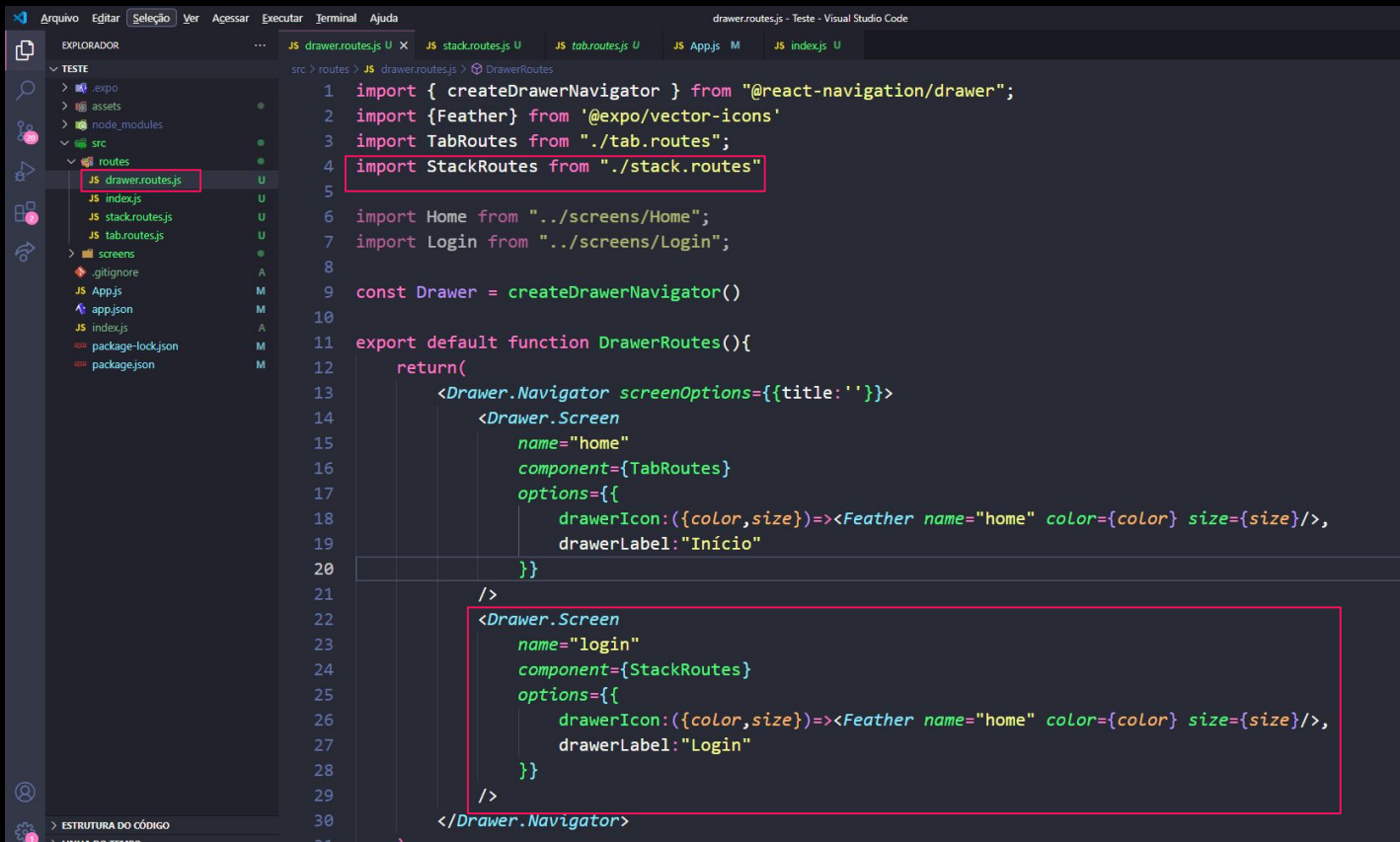


The image shows a screenshot of the Visual Studio Code editor interface. On the left, the 'EXPLORADOR' (Explorer) sidebar is open, showing a file tree for a project. The tree includes folders like '.expo', 'assets', 'node_modules', and 'src'. Under 'src', there is a 'routes' folder containing 'drawer.routes.js', 'index.js', 'stack.routes.js' (which is selected and highlighted), and 'tab.routes.js'. Other files in the project include 'App.js', 'app.json', 'index.js', 'package-lock.json', and 'package.json'. The main editor area displays the content of 'stack.routes.js'. The code defines a 'DrawerRoutes' function that returns a configuration for a 'Stack.Navigator'. It imports 'createNativeStackNavigator' from '@react-navigation/native-stack', 'Feather' from '@expo/vector-icons', and 'Login' from '../screens/Login'. The navigator is configured with a single screen named 'home' that uses the 'Login' component.

```
1 import { createNativeStackNavigator } from '@react-navigation/native-stack';
2 import { Feather } from '@expo/vector-icons'
3 import TabRoutes from "../tab.routes";
4 import Home from "../screens/Home";
5 import Login from "../screens/Login";
6
7 const Stack = createNativeStackNavigator()
8
9 export default function DrawerRoutes(){
10   return(
11     <Stack.Navigator screenOptions={{title:''}}>
12       <Stack.Screen
13         name="home"
14         component={Login}
15       />
16     </Stack.Navigator>
17   )
18 }
```

Importando a navegação stack

FIA/P



The screenshot shows the Visual Studio Code editor with the file explorer on the left and the code editor on the right. The file explorer shows the project structure with the following files and folders:

- EXPLORADOR
 - TESTE
 - .expo
 - assets
 - node_modules
 - src
 - routes
 - JS drawer.routes.js (selected)
 - JS index.js
 - JS stack.routes.js
 - JS tab.routes.js
 - screens
 - .gitignore
 - JS App.js
 - app.json
 - index.js
 - package-lock.json
 - package.json

The code editor shows the content of `drawer.routes.js`:

```
1 import { createDrawerNavigator } from "@react-navigation/drawer";
2 import { Feather } from '@expo/vector-icons'
3 import TabRoutes from "../tab.routes";
4 import StackRoutes from "../stack.routes"
5
6 import Home from "../screens/Home";
7 import Login from "../screens/Login";
8
9 const Drawer = createDrawerNavigator()
10
11 export default function DrawerRoutes(){
12   return(
13     <Drawer.Navigator screenOptions={{title:''}}>
14       <Drawer.Screen
15         name="home"
16         component={TabRoutes}
17         options={{
18           drawerIcon: ({color, size}) => <Feather name="home" color={color} size={size}/>,
19           drawerLabel: "Início"
20         }}
21       />
22       <Drawer.Screen
23         name="login"
24         component={StackRoutes}
25         options={{
26           drawerIcon: ({color, size}) => <Feather name="home" color={color} size={size}/>,
27           drawerLabel: "Login"
28         }}
29       />
30     </Drawer.Navigator>
31   )
32 }
```

Dúvidas?